

САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
КАФЕДРА КОМПЬЮТЕРНОГО МОДЕЛИРОВАНИЯ И
МНОГОПРОЦЕССОРНЫХ СИСТЕМ

Закуанова Маргарита Ринатовна

Выпускная квалификационная работа бакалавра

**3D-реконструкция модели помещения
по его изображениям**

Направление 010300

Фундаментальная информатика и информационные технологии

Научный руководитель,
кандидат тех. наук,
доцент
Гришкин В. М.

Санкт-Петербург

2017

Содержание

Введение	3
Постановка задачи	5
Обзор литературы	7
Глава 1. Обзор существующих решений	9
1.1. 3D-реконструкция с помощью проектирования.....	9
1.2. 3D-реконструкция с помощью набора изображений объекта со всех сторон.....	11
1.3. 3D-реконструкция с помощью 3D-сканера.....	12
1.4. 3D-реконструкция с помощью стереопары	13
1.5. Выводы.....	14
Глава 2. Этапы работы алгоритма построения 3D-модели по стереопаре.....	15
2.1. Взаимное расположение камер при создании изображений....	15
2.2. Поиск ключевых точек.....	17
2.2.1. Алгоритм Speeded-Up Robust Features (SURF).....	17
2.2.2. Алгоритм Oriented FAST and Rotated BRIEF (ORB).....	19
2.3. Сопоставление и фильтрация «плохих» пар особых точек.....	20
2.3.1. Сопоставление особых точек	20
2.3.2. Фильтрация «плохих» пар особых точек	21
2.4. Ректификация стереопары, получение карты глубины	23
2.5. Выводы.....	25
Глава 3. Практическая реализация	26
3.1. Выбор языка программирования и инструментов разработки	26
3.2. Реализация выбранного метода	27
3.3. Тестирование работы программы	27
3.4. Выводы	28
Заключение	30
Список литературы	31

Введение

В современном мире большое внимание уделяется вопросам развития информационных технологий и оборудования в IT-индустрии. Ежегодно разработчики демонстрируют улучшенные версии как программных продуктов, так и новейших технических устройств, которые постепенно становятся все более доступными, ввиду конкурентной борьбы между производителями за конечных потребителей.

На сегодняшний день в различных технических устройствах для связи значительному большинству пользователей доступна мобильная камера. К услугам камеры пользователи прибегают все чаще и чаще. И, как следствие, расширяются масштабы использования фотографий в нашей повседневной жизни: человек старается запечатлеть все происходящее вокруг себя – от фиксации необходимых документов до пополнения коллекции изумительных вечерних закатов. Отчасти на это повлияло и развитие Интернета: люди стремятся не только сохранить в памяти значимые моменты их жизни, но и поделиться ими со своими друзьями в социальных сетях.

Активное пополнение пользователями своих фото- и видеоархивов, неизбежным образом повышает актуальность разработок с использованием новейших технологий устройств хранения данных. В XXI веке начинают использовать накопители с флеш-памятью, превосходящие «устаревшие» дискеты не только по своим компактным габаритам, но и по многим другим техническим характеристикам.

С другой стороны, ежедневно порождаемые десятки и сотни терабайт информации требуют грамотной обработки и продуманной структуризации. Решением данного вопроса озадачено не одно направление информационных технологий. В их числе и, так называемое, компьютерное зрение.

В различных сферах человеческой деятельности присутствуют системы, использующие компьютерное зрение. Огромную роль данные системы играют в медицине, помогая находить опухоли и другие патологии,

в результате анализа снимков больных [1]; на производстве, например, в пищевой промышленности, оценивая качество и минимизируя человеческий фактор. Также их можно использовать и в познавательных целях. В настоящее время огромной популярностью пользуются онлайн технологии, благодаря которым человек может дистанционно наблюдать и изучать желаемый объект, даже будучи на больших расстояниях от объекта исследования. Существует целый ряд приложений по анализу трехмерных изображений и структур.

На данный момент одними из самых популярных библиотек для работы с алгоритмами компьютерного зрения являются OpenCV (Open Source Computer Vision Library) и PCL (Point Cloud Library). Библиотеки кроссплатформенные, реализованы на языке высокого уровня(C/C++), но разрабатываемые и для других языков программирования. Распространяются в условиях лицензии BSD, следовательно, могут быть использованы как в научных, так и в коммерческих целях.

Постановка задачи

Целью данной работы является построение 3D-модели помещения на основе его изображений.

Для достижения поставленной цели осуществлена разработка прототипа программного обеспечения, использующего библиотеку OpenCV, в рамках которого решаются следующие задачи:

1. Нахождение взаимного расположения камер при создании изображений;
2. Поиск особых точек:
 - Анализ алгоритма Speeded-Up Robust Features (SURF);
 - Анализ алгоритма Oriented FAST and Rotated BRIEF (ORB);
3. Анализ методик сопоставления ключевых точек в пары с последующей фильтрацией «плохих» пар:
 - Перекрестная проверка;
 - Проверка с RANSAC;
4. Ректификация стереопары, получение карты глубины.

Введем некоторые определения и сокращения:

1. Трёхмерная модель – математическая модель предмета, отражающая его размеры и свойства с определенной точностью. Представляет из себя множество точек, расположенных в трехмерной системе координат.
2. Стереозрение – получение трехмерной картины мира по видеоряду или нескольким изображениям.
3. Стереопара – две камеры, расположенные по определенным правилам, при обработке данных с которых можно получить стереоизображение (в некоторых случаях имеет значение пары изображений, полученных с подобных камер).
4. Стереобазис – расстояние между объективами в стереопаре.

5. Карта глубин – изображение, в котором для каждого пикселя содержится информация о расстоянии между объектом и камерой.
6. Соответствующие точки – точки на изображениях, которые на реальном объекте расположены на одном и том же месте.
7. Эпиполярная геометрия – раздел геометрии, описывающий проецирование геометрических объектов в пространстве.
8. Ректефикация – в данном случае, вычисление координат особых точек в единой системе координат.

Обзор литературы

Компьютерное зрение можно охарактеризовать как достаточно молодое направление, активно развивающееся с 1970-х годов. На сегодняшний день издано множество книг и статей, касающихся различных областей компьютерного зрения и обработки изображений.

В книгах [2, 4] зарубежных авторов собрана информация, позволяющая полностью погрузиться в данное направление. По данной литературе можно ознакомиться с основными понятиями компьютерного зрения, с распространенными задачами, возникающими в данной области и с алгоритмами их решения. Также существуют подобные издания, переведенные на русский язык, например, [3]. Большой популярностью пользуются и видео-уроки [5, 6, 7], содержащие аналогичную информацию.

Задача построения 3D-модели подробнее освещается в источниках [8, 9, 10]. Однако, здесь авторы рассматривают различные способы реконструкции некоторых конкретных объектов: тело человека, памятник, посуда, книга, не затрагивая реконструкцию окружающей среды. Способ реконструкции основанный на стереопарах освещается в статьях [11, 12].

Одним из этапов решение задачи данной исследовательской работы является поиск ключевых точек. Tuytelaars T., Mikolajczyk K. в своем обзоре [13] рассматривают детекторы особых точек, их эволюцию с течением времени, анализ работы, и каковы их достоинства и недостатки. В статьях [16, 17] особое внимание уделяется методу ORB, как альтернативе предшествующим SIFT и SURF.

Задача сопоставления особых точек и фильтрации неподходящих пар освещается в указанных выше источниках [2, 4]. Martin A. Fischler и Robert C. Bolles в своей книге [19] упоминают алгоритм RANSAC и его применение в работе с изображениями.

Далее разработка прототипа программного обеспечения реализуется с помощью библиотеки OpenCV. В данных источниках [20, 21, 22] теорию

компьютерного зрения рассматривают через призму программной реализации, то есть здесь можно найти информацию, касающуюся как начала работы с данной библиотекой, так и реализации конкретных алгоритмов.

Также для построения 3D-модели может быть использована библиотека PCL, обзор которой приводится в статье [23]. Irene Reisner-Kollmann в своей диссертации [24], проводит обзор и оценку различных методов построения 3D-модели исключительно в рамках данной библиотеки.

Глава 1. Обзор существующих решений

На сегодняшний день предлагается целый ряд решений задачи построения 3D-модели объектов. Некоторые из них предполагают использование специального оборудования, например, 3D-сканеров, другие же могут ограничиться возможностями веб-камер. Основная часть методов заключается в нахождении расположения точек объекта, их координат в координатном пространстве без изменения масштаба, и по необходимости, в наложении текстур на точки модели [2].

Для получения результата высокого качества вне зависимости от выбранного метода (за некоторым исключением), необходимо выполнение некоторых условий [11]:

1. Высокое качество входных данных;
2. Одинаковое освещение поверхностей при изменении угла обзора;
3. Калибровка камер (изучение их технических параметров и расположения в пространстве);
4. Существование областей пересечений у изображений;
5. Резкая смена интенсивности на границе объекта.

1.1. 3D-реконструкция с помощью проектирования

Данный метод предполагает разработку образа желаемого объекта вручную. Существует целый ряд программ, использующих данный метод, причем как в офлайн (Blender, Autodesk 3D max) (рис. 1), так и в онлайн (Tinkercad, Sketchfab) (рис. 2) вариантах. При создании модели объекта проектировщик строит его образ на основе своего восприятия, прямое взаимодействие с изображениями объекта отсутствует. Таким образом с помощью данного метода можно строить 3D-модели как конкретных объектов (здания, мебель), так и объектов, существующих только в воображении проектировщика (различные абстракции).

Этапы построения 3D-модели:

1. Построение полигональной модели, называемой каркасом. Достигается путем восстановления геометрической формы и размеров желаемого объекта;
2. Выбор составляющего материала. Например, применение к зданию при его проектировании текстуры дерева;
3. Настройка освещения;
4. Настройка сцены: расположение камер для наблюдателя.
5. Создание внешних воздействий (при необходимости).

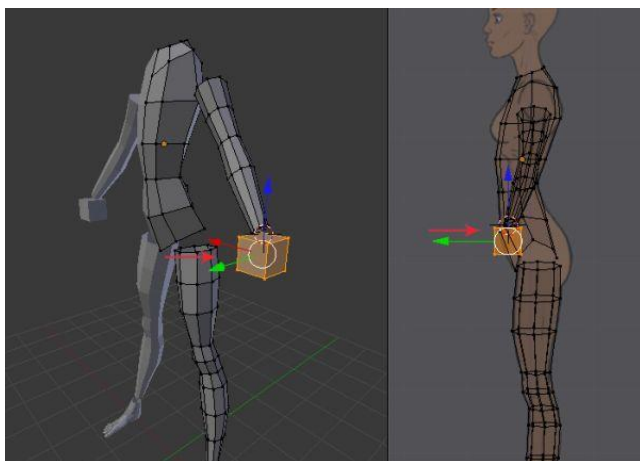
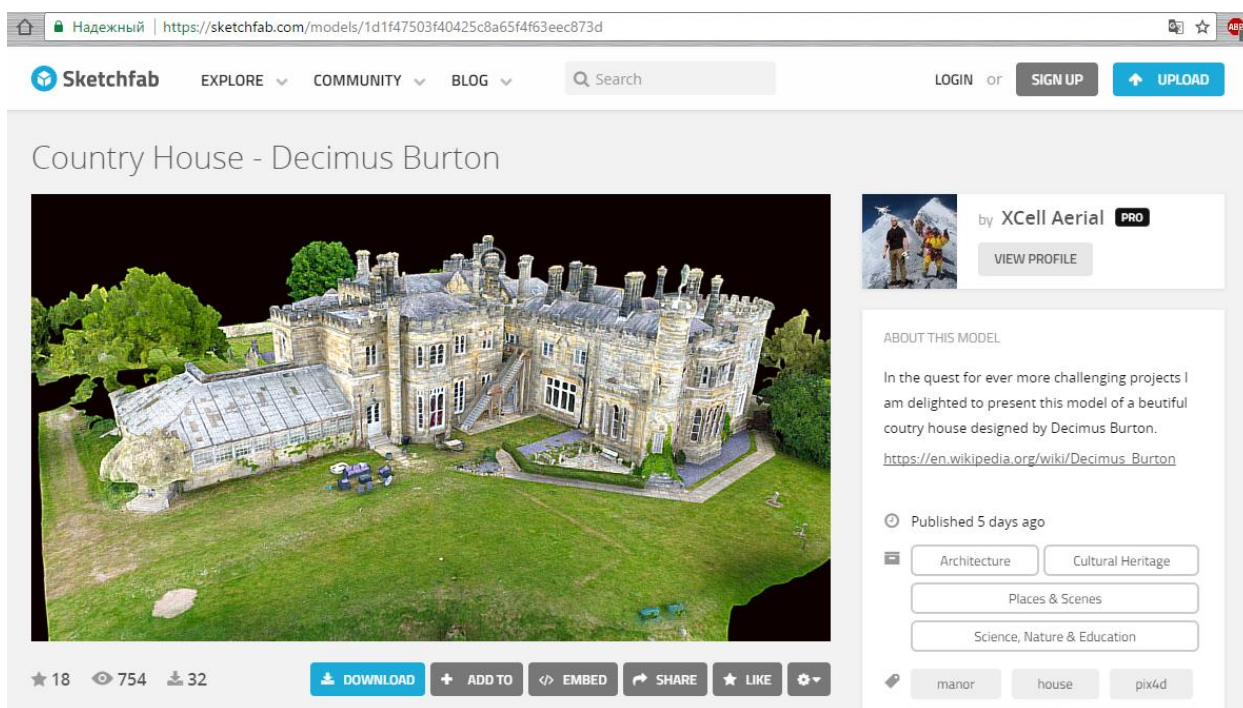


Рис 1. Процесс моделирования тела человека с использованием Blender

Рис. 2. Процесс моделирования архитектуры с использованием Sketchfab



Данный метод имеет множество преимуществ. Он обладает высокой точностью, что позволяет проектировщику через отражение малейших деталей и размеров осуществлять построение моделей любых объектов. Однако, для получения удовлетворительного результата требуется много времени, что, несомненно, является существенным недостатком метода.

1.2. 3D-реконструкция с помощью набора изображений объекта со всех сторон

Метод использует последовательный ряд изображений объекта (рис. 3). При этом требуемый процент наложения двух смежных кадров должен превышать 50%, а минимальное количество перекрывающихся кадров должно быть равно трем. При грамотном выполнении этих условий можно получить достаточно качественный образ, с последующим редактированием лишь размера объекта [8, 9].

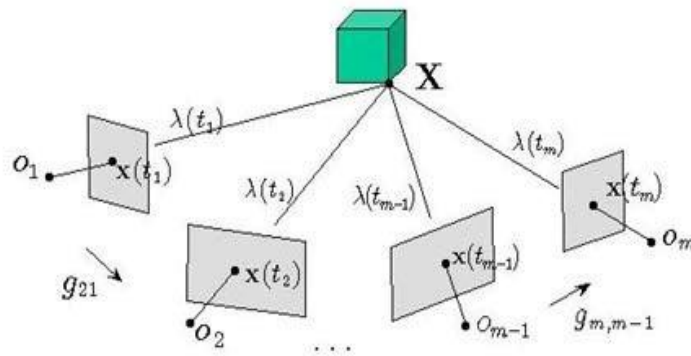


Рис. 3. Общая схема фотографирования объекта

Алгоритм работы данного метода можно представить следующим перечнем этапов [10]:

1. Фотографирование исследуемого объекта и импорт полученных изображений;
2. Поиск особых точек и решение системы уравнений, полученной на основании множества данных точек;
3. Определение параметров камеры;
4. Поиск «одинаковых» точек на различных наборах смежных изображений объекта;

5. Вычисление координат точек относительно «базового» изображения объекта;
6. Приведение точек к системе координат, наиболее удобной для анализа объекта и наложения структуры.

В данном методе в качестве недостатков можно выделить стационарное положение объекта и долгий процесс съемки. Однако, увеличение количества фотографий приводит к более качественным результатам.

1.3. 3D-реконструкция с помощью 3D-сканера

3D-сканер – периферийное устройство, анализирующее объект и на основе полученных данных создающее его 3D-модель.

По методу сканирования они делятся на два типа: контактный и бесконтактный. Первый предполагает использование специального щупа, который исследует точки, выбранные оператором. Точность получаемой информации, зависит от датчиков, расположенных на щупе, которые схожи с датчиками оптико-механической мыши. Данный тип сканеров предполагает работу с объектами небольшого размера и простой формы. Бесконтактные 3D-сканеры имеют более сложную структуру и оперируют ультразвуком, различными лазерными датчиками и фотокамерами. Происходит комбинирование данных самих изображений и результатов, полученных от других используемых датчиков.

Ярким примером служит сканирование, использующее белый свет (рис. 4). Сканер направляет на объект шаблон закодированного света от источника, который и «присваивает» форму объекту. Данные согласовываются со снимками и образуют качественную модель.

Данный метод позволяет получить самую точную 3D-модель за короткий промежуток времени. Однако данные технические установки не только дорогостоящие, но и стационарные, что не позволяет реконструировать любой объект.

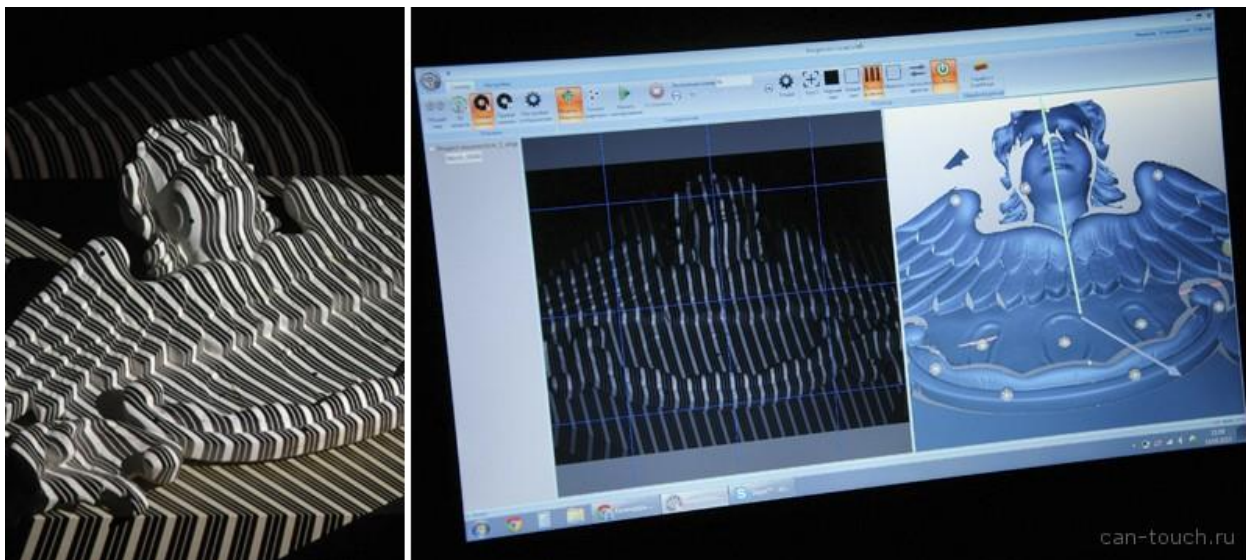


Рис. 4. Сканирование белым светом

1.4. 3D-реконструкция с помощью стереопары

Метод оперирует набором лишь из двух изображений, полученных с помощью стереокамеры и является, можно сказать, подзадачей технологии, описанной в п.1.2. В данном случае аналогично осуществляется выбор точек соответствия, их сопоставление и геометрические преобразования [10, 11].

Таким образом, для получения 3D-модели необходимо выполнение следующего алгоритма действий:

1. Определение фундаментальной матрицы;
2. Определение матрицы камер;
3. Нахождение соответствующих точек;
4. Построение облака точек;
5. Текстурирование.

Однако, модель, полученная подобным образом, не считается «полным» 3D-образом, так как строится лишь поверхностный «облик» объекта. Привлекает же к себе данный метод своими быстротой и доступностью.

1.5. Выводы

Таким образом, на основе проведенного анализа можно утверждать, что задача 3D-реконструкции очень актуальна и существует множество способов ее решения в зависимости от поставленных ограничений. Для решения задачи данной работы был выбран метод, использующий стереопары.

Глава 2. Этапы работы алгоритма построения 3D-модели по стереопаре

В предыдущей главе был приведен краткий обзор различных методов, решающих задачу 3D-реконструкции. В данной главе подробнее остановимся на методе, выбранном для решения поставленной задачи.

2.1. Взаимное расположение камер при создании изображений

Как упоминалось в предыдущей главе, на изображения накладывается ряд ограничений, что увеличивает значимость вопроса о расположении камер.

Изображения исследуемого объекта можно получить двумя основными способами съемки: параллельным (рис.5) или направленным (рис.6).

При использовании первого метода съемки оптическая ось объективов источников не меняется. Однако, при дальнейшей обработке изображений требуется использование компенсационного сдвига, посредством которого достигается нулевой параллакс на одном из объективов. В данном случае стереопарой будет являться лишь та область изображения, которая попадает в область пересечений. Данный метод пользуется популярностью ввиду своей простоты. Еще одним преимуществом является то, что при получении стереопар подобным образом, смещение будет наблюдаться лишь у объектов на переднем плане, а далеко расположенные объекты останутся на своих местах.

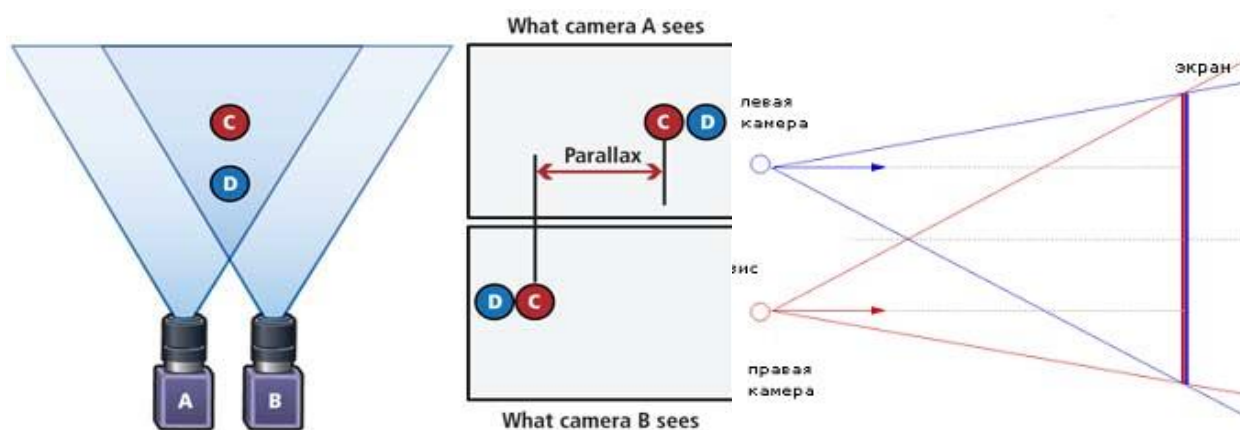


Рис. 5. Параллельный способ съемки объекта

При направленном методе съемки поворот камеры осуществляется таким образом, чтобы объектив всегда был направлен на объект фокусировки, что в последующем не требует обработки полученных результатов. Используя этот метод, возможно получение модели наиболее близкой к естественному восприятию. Но при этом, к возникающим трудностям можно отнести сложность процесса съемки объекта.

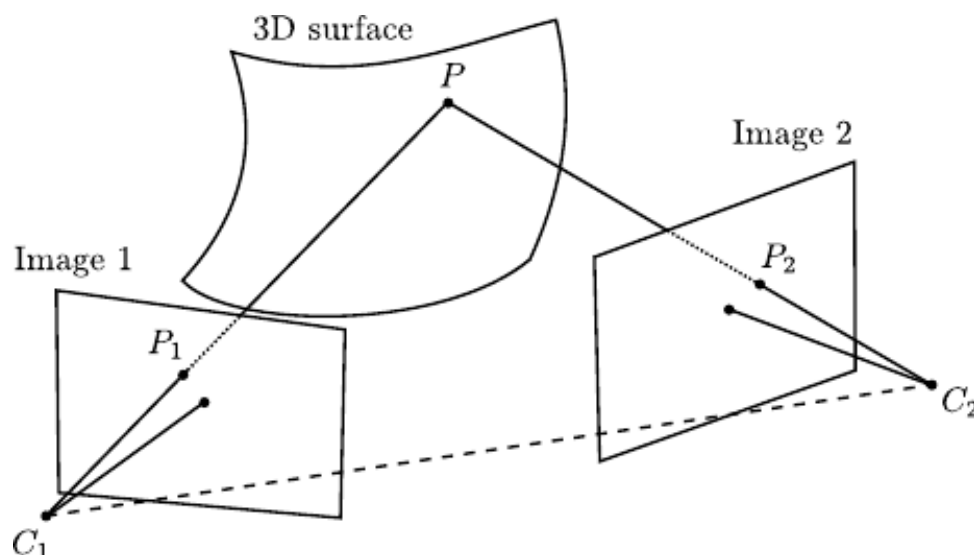


Рис. 6. Направленный способ съемки объекта

Для получения качественного результата необходимо:

1. Знать фокусные расстояния объективов и расстояния между ними;
2. Обеспечить параллельность осей и совпадение плоскостей кадра;
3. Синхронизировать работу затворов камер в случае присутствия движущегося объекта.

В таблице 1 приведена зависимость расстояния между объектом и камерой (d) и стереобазисом (b), которая может быть выражена формулой:

$$b = 0,03 * d$$

где b – базис стереосъемки, d – расстояние до ближайшего объекта фотографируемой сцены.

Используемый в формуле коэффициент (0,03) получен из угла схождения зрительных осей, обеспечивающего наилучшие условия для наблюдения стереоэффекта.

Расстояние, d (м)	Стереобазис, b (см)
1-2	6-9
2-3	9-12
3-4	12-15
4-5	15-18
5-6	18-21
6-7	21-24

Табл. 1. Соотношение расстояния между объектом и камерой (d) и стереобазисом (b)

2.2. Поиск ключевых точек

Для сопоставления изображений с целью поиска областей пересечений, необходимо выделить характеризующие признаки – особые (ключевые) точки. Под такими точками подразумевают точки, удовлетворяющие следующим свойствам: определенность, устойчивость, инвариантность, стабильность, интерпретируемость. Данные признаки обеспечивают преимущества алгоритмов, основанных на поиске ключевых точек [13].

Метод работы с изображениями на основе ключевых точек довольно популярен. Результатом его использования служит набор параметров в виде числовых или бинарных векторов, определяющих окрестности ключевых точек – дескрипторов. Далее, при анализе изображений происходит сравнение дескрипторов с использованием определенной метрики.

На основе изученных публикаций [16, 17], и с учетом поставленных требований, были выбраны два метода, реализованные в библиотеке OpenCV. Рассмотрим каждый из них подробнее.

2.2.1. Алгоритм Speeded-Up Robust Features (SURF)

SURF решает задачу поиска особых точек и создания их дескрипторов, инвариантных к масштабу и вращению – при изменении масштаба или поворотах изображения, описания ключевых точек не изменяются.

Метод осуществляет поиск особых точек с помощью матрицы Гессе (1). Ее определитель достигает экстремума в точках максимального изменения градиента яркости. Исходя из этого, получаем, что такие элементы изображений как пятна, углы и края линий будут достаточно хорошо «узнаваемы».

$$Gs(f(x, y)) = \begin{bmatrix} \frac{\partial^2 f}{\partial x^2} & \frac{\partial^2 f}{\partial x \partial y} \\ \frac{\partial^2 f}{\partial x \partial y} & \frac{\partial^2 f}{\partial y^2} \end{bmatrix}, \quad \det(Gs) = \frac{\partial^2 f}{\partial x^2} \frac{\partial^2 f}{\partial y^2} - \left(\frac{\partial^2 f}{\partial x \partial y} \right)^2, \quad (1)$$

где Gs – матрица Гессе, $f(x, y)$ – функция изменения градиента яркости

Инвариантность масштабу SURF обеспечивается использованием разномасштабных фильтров при нахождении гессианов. Для каждой особой точки рассчитывается масштаб и направление максимального изменения яркости, формируется дескриптор, представляющий собой набор из 64 (либо 128) чисел.

Для обеспечения эффективности вычислений каждое изображение, перед «запуском» алгоритма, представляется в интегральном формате. Данный формат предполагает использование матрицы, размерность которой эквивалентна размеру самого изображения. Элементы же матрицы вычисляются по формуле (2).

$$I(x, y) = \sum_{i=0}^x \sum_{j=0}^y I(i, j), \quad (2)$$

где $I(i, j)$ — яркость пикселя входного изображения, (x, y) — координаты пикселя на матрице

Алгоритм SURF более эффективен при анализе изображений с размытием и наличием вращений. Обработка же изображений с сильным изменением освещенности создаст определенные трудности для данного алгоритма. Также метод не «сработает» при обработке простых (с однотонной текстурой) объектов. SURF воспринимает картинку как

единое целое и в таких условиях осуществляет поиск его особенностей. Точки будут обнаружены на границе объекта с фоном и при изменении объекта возникнут проблемы с его идентификацией.

Так же, недостатком данного алгоритма может служить наличие патентных ограничений, что, некоторым образом, ограничивает возможности его использования.

2.2.2. Алгоритм Oriented FAST and Rotated BRIEF (ORB)

Алгоритм ORB опубликован в 2011г [16, 17]. В его основе лежит комбинация таких алгоритмов, как детектор FAST (**F**eatures from **A**ccelerated **S**egment **T**est) и дескриптор BRIEF (**B**inary **R**obust **I**ndependent **E**lementary **F**eatures) с некоторыми улучшениями.

Для нахождения особых точек используется детектор FAST, который неоднократно демонстрировал себя в качестве быстрого инструмента при обнаружении особенностей изображения. Метод FAST не инвариантен к поворотам, а потому, возникает необходимость ввода параметра угловой ориентации. Указанный параметр обеспечивает детектирование при вращении объекта, использует направление с наибольшей интенсивностью, назначаемое ориентацией особой точки θ .

Дескриптор BRIEF, представляющий собой вектор длиной, равной 256, состоит из результатов бинарных тестов вокруг особой точки. Для достижения инвариантности к вращению область вычисления дескриптора определяется ориентацией особой точки θ . Дескриптор BRIEF обеспечивает распознавание объекта с разных ракурсов при достаточно малых затратах на вычисления.

Скорость работы алгоритма ORB выше, чем у алгоритма SURF, так как используются не все особые точки, а только та их часть, которая остается после «отсева» детектора углов Харриса.

Еще одним преимуществом данного алгоритма является его публикация под BSD лицензией в библиотеке OpenCV.

В некоторых задачах ORB позволяет обнаруживать особенности изображений в режиме реального времени. Подход к поиску ключевых точек и их описаний, свойственный ORB, позволяет использовать его на менее мощных вычислительных устройствах.

2.3. Сопоставление и фильтрация «плохих» пар особых точек

По итогам работы алгоритмов, описанных в предыдущей главе, мы получаем множество дескрипторов, описывающих особые точки изображения.

Далее необходимо найти соответствие между особыми точками, с целью нахождения «похожих», удалить из рассмотрения «плохие» (неподходящие) пары, чтобы в дальнейшем последние не повлияли на процесс генерирования панорамы помещения.

2.3.1. Сопоставление особых точек

Для решения данной задачи в OpenCV существует несколько методов, среди которых BruteForceMatcher и FlannBasedMatcher [20].

По наименованиям указанных методов можно видеть, что первый из них выполняет полный перебор различных комбинаций дескрипторов с изучаемых изображений. Данный метод осуществляет поиск для дескриптора с одного изображения «ближайшего» соседа-дескриптора с другого, вычисляя при этом расстояния между ними. Имея большой объем обрабатываемой информации, специфика алгоритма BruteForceMatcher выступает в качестве недостатка, так как потребует значительное время для сопоставления особых точек.

Для решения данной проблемы существует метод FlannBasedMatcher, базирующийся на библиотеке Fast Library for Approximate Nearest Neighbors (FLANN). Библиотека содержит множество алгоритмов для извлечения и вычисления дескрипторов, индексации и применяемый в данном методе поиск KNN или метод поиска k-ближайших соседей.

По итогам второй главы для выявления особых точек был выбран алгоритм ORB, в результате работы которого мы получаем бинарные дескрипторы конечного числа точек. При решении задачи рассматривается умеренное количество данных, в связи с чем было решено процедуру сопоставления осуществлять путем перебора. Как мы уже знаем, для сравнения дескрипторов ORB необходимо использование специальной метрики. В данном случае используется расстояние Хэмминга, которое определяет количество различных позиций между двумя бинарными последовательностями. Такую метрику можно представить в виде формулы (3).

$$hamming(x, y) = \sum_{x_i \neq y_i} 1, \quad i = 1, \dots, n, \quad (3)$$

где x и y – дескрипторы

2.3.2. Фильтрация «плохих» пар особых точек

Методы сопоставления точек, описанные в предыдущем параграфе, в процессе работы «совершают» ошибки. Происходит объединение особых точек, на самом деле не являющихся отражением одного и того же объекта. Некорректные связи сильно влияют на результат. Например, при построении панорамы мы можем в результате получить неверное объединение фрагментов.

Решение этой проблемы имеет большое значение, и на сегодняшний день существует немало алгоритмов фильтрации. Эти алгоритмы можно разделить на две группы: простые и итеративные фильтры. Рассмотрим некоторые фильтры, не требующие при их реализации значительных временных ресурсов.

Метод перекрестной проверки. Рассмотрим ситуацию с анализом двух изображений. С помощью дескриптора мы находим пару точек («1» и «2») и устанавливаем между ними соответствие. Если мы предполагаем, что это отношение истинно, то существует в некотором роде взаимно-однозначное соответствие. Другими словами, если к точке «1» самой

ближайшей оказалась точка «2», то и для точки «2» ближайшей должна оказаться точка «1».

Это поможет нам исключить связи, в которых особой точке на одном изображении соответствует несколько точек на другом изображении. Данный алгоритм не самый быстрый в работе и, кроме того, возможны ситуации, при которых ошибочно отбрасываются истинные связи. Несмотря на это, алгоритм способен демонстрировать эффективность при фильтрации большого количества ложных связей.

Алгоритм RANSAC. RANSAC (RANdom Sample And Consensus) – один из наиболее популярных в различных областях обработки изображений. Он не стоит на месте и постоянно совершенствуется разработчиками. Предлагаются новые решения, превосходящие по скорости и качеству изначальные версии алгоритма [19].

Данный метод относится к группе итеративных, и каждый его шаг можно разбить на этапы:

1. Выбор случайного количества тестируемых точек и вычисление модели на их основе;
2. Для вычисленной модели выполняется классификация других точек (не «участвующих» в первом этапе), считается значение потерь, и если полученный результат превосходит наилучший сохраненный ранее, данный набор допустимых точек запоминается для дальнейших проверок.

Для продуктивного использования алгоритма RANSAC, необходимо грамотно выбирать количество итераций алгоритма. При выборе небольшого количества итераций, мы, возможно, не доберемся до наилучшей модели, а при большом его количестве сильно «потеряем» во времени.

2.4. Ректификация стереопары, получение карты глубины

После получения особых точек необходимо найти их координаты в единой системе, и далее проверить полученные пары точек на выполнение условия стереопары. Суть данного условия будет описана ниже.

Для начала опишем некоторые геометрические понятия и свойства. Рассмотрим две камеры (рис.7). C, C' – их центры. Точка пространства X проецируется на плоскости изображений левой и правой камер в точки x и x' соответственно. Прообразом точки x является луч, исходящий от точки пространства X . Этот луч проецируется на плоскость второй камеры в прямую l' , называемую эпиполярной линией. Образ точки X на плоскости изображения второй камеры обязательно лежит на эпиполярной линии l' .

Таким образом, существует взаимно-однозначное соответствие между каждой точкой одного изображения и эпиполярной линией второго изображения. При этом парная точка для x на изображении правой камеры может лежать только на соответствующей эпиполярной линии.

Эпиполярная геометрия имеет очень простую запись в координатах. Существует такая матрица F размера 3×3 , что пара точек x и x' является стереопарой тогда и только тогда, когда выполняется равенство:

$$x'^T F x = 0$$

Матрица F называется фундаментальной матрицей (fundamental matrix). С ее помощью вычисляются уравнения эпиполярных линий. Для точки x уравнение будет иметь вид: $l'^T x' = 0$. Ранг F равен двум, она определена с точностью до ненулевого множителя и зависит только от матриц исходных камер P и P' .

Так же важным понятием в данной области является существенная матрица (essential matrix): $E = K^T F K$. В случае, когда K, K' – матрицы внутренних параметров камеры будут единичными, данная матрица совпадет с фундаментальной. По ней можно определить положение и поворот второй

камеры относительно первой, что используется в задачах определения движения камеры.

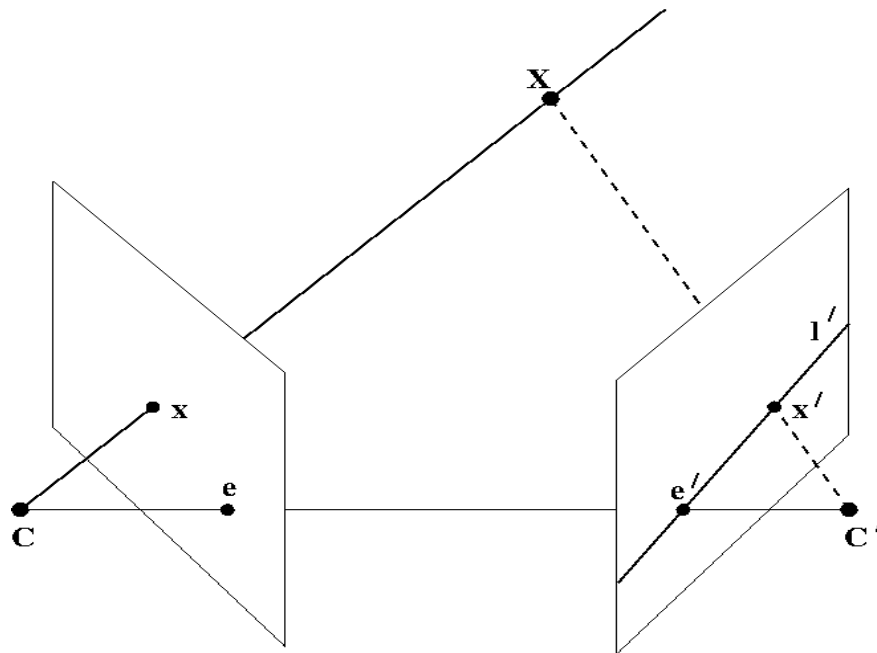


Рис.7. Эпиполярная геометрия

Теперь обсудим вопрос получения трехмерных координат точки по координатам ее проекций, то есть процесс триангуляции. Он сводится к решению следующей системы:

$$\begin{cases} x = PX \\ x' = P'X \end{cases}$$

Используя вычисленные координаты, мы получаем координаты прообраза. Таким образом, мы можем легко определить расстояния до плоскости камеры, а значит и возможность получить карту глубины (рис.8). Также ее можно оперативно получить с помощью специального оборудования, например, сенсора Kinect.



Рис. 8: Пример ректифицированных картинок, и соответствующей им карты глубины

2.5. Выводы

Таким образом, для построения 3D-модели необходимо выполнение этапов, перечисленных в данной главе. А на некоторых из них, выбрать определенные методы решения в зависимости от конкретной задачи и входных данных. Результатом будет являться карта глубин, которую можно считать 3D-реконструкцией. Для достижения лучшего визуального представления, можно воспользоваться сопоставлением карты глубины и изображений.

Глава 3. Практическая реализация

В данной главе будут описаны средства, с помощью которых решалась поставленная задача, и представлены полученные результаты.

3.1. Выбор языка программирования и инструментов разработки

Для работы с изображениями, как уже упоминалось в предыдущих главах, ввиду наличия множества алгоритмов компьютерного зрения и обработки изображений была выбрана библиотека OpenCV версии 3.1.

Для разработки программы выбор был сделан в пользу объектно-ориентированного языка C++. Последний обладает рядом следующих преимуществ:

1. Масштабируемость (данный язык используют в приложениях для множества различных платформ и систем);
2. Соответствие объектно-ориентированному подходу;
3. Популярность (множество примеров реализации различных алгоритмов, информация о решениях, возникающих проблем);
4. Язык разработки библиотеки OpenCV.

Так же использовалась интегрированная среда разработки Microsoft Visual Studio 2015, имеющая множество инструментов для создания программного обеспечения. В данной среде можно создавать как простые приложения, так и сложные системы. Visual Studio обладает устойчивой работоспособностью и интегрируется с другими приложениями.

3.2. Реализация выбранного метода

В качестве первого шага, средствами встроенного в OpenCV модуля `calib3d` (`RunCalibrationAndSave`), производится калибровка используемых для съемки объекта камер [21].

В результате калибровки мы получаем параметры модели камеры в виде матриц, которые ранее упоминались в параграфе 2.4 (P , K).

В качестве входных параметров программного модуля выступают два изображения объекта. Предполагается, что данные изображения удовлетворяют критериям стереопары, обладают удовлетворительным разрешением при отсутствии помех. Предполагается также статичность камер и наличие верно определенного стереобазиса.

Для выявления особых точек на основе анализа, приведенного в параграфе 2.2, был выбран алгоритм Oriented FAST and Rotated BRIEF (`cv2.ORB_create()`), дескрипторы которого в дальнейшем сравниваются методом полного перебора (`cv2.BFMatcher`) с параметром RANSAC, определяющим фильтрацию [22].

На заключительном этапе с помощью математических функций на базе матриц, полученных на первом шаге, производится анализ особых точек с последующим вычислением их координат. Результатом служит вывод карты глубины модели объекта.

3.3. Тестирование работы программы

Для анализа результата работы разработанной программы, были сформированы 20 стереопар различных объектов. Каждая из которых отличалась своими параметрами, например, количеством объектов на изображении или расстоянием от камеры до объекта.

По оценке эксперта, по 5 стереопарам получен «отличный» результат, по 6 – «хороший», по 5 – «удовлетворительный», и по оставшимся – «неудовлетворительный».

На основании полученных данных можно сделать вывод о том, что наилучший результат достигается при небольшом количестве объектов на фотографии. Также статистика показывает, что увеличение расстояния до объекта нехорошо сказывается на качестве полученного результата.

Примеры полученных карт глубин для исходных стереопар можно увидеть на рис. 9.

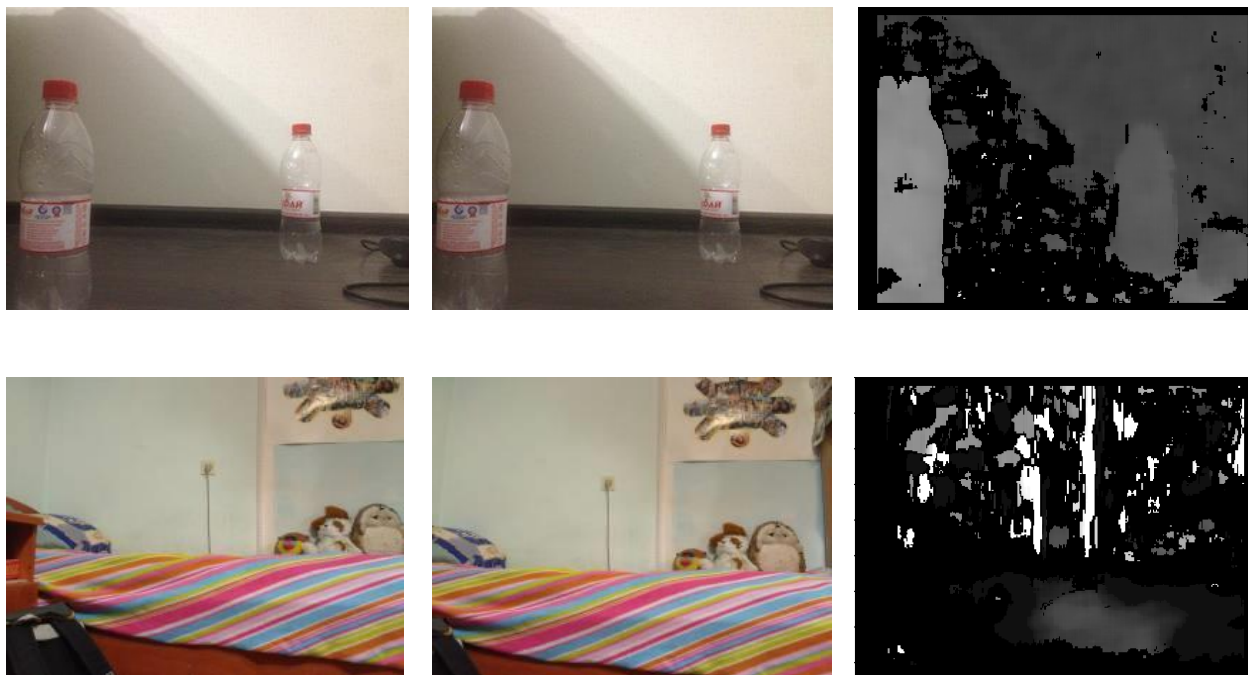


Рис.9: Стереопара и полученная по ней карта глубины

3.4. Выводы

Таким образом, на основе изученного материала был реализован прототип программного обеспечения на языке C++, с использованием библиотеки OpenCV 3.1, который осуществляет построение карты глубины по стереопаре. В своем роде данный результат можно назвать 3D-моделью, так как он дает информацию о координатах точек в трехмерном пространстве. Однако данный прототип может быть улучшен путем текстурирования и на основе этих данных получена реалистичная 3D-модель помещения.

В проведенном исследовании в качестве объектов выступают фрагменты помещений. Построенные карты глубин отражают как элементы внутреннего интерьера помещений (витрины, стеллажи), так и пространственные характеристики (стены, углы, выступы) контуров самих помещений. При грамотном объединении набора карт глубин указанных фрагментов, можно получить цельную реконструкцию помещений.

Заключение

Наш век называют веком информационных технологий. На сегодняшний день активно используется Интернет для разных целей: развлекательных, образовательных, коммуникационных. У человека не всегда есть возможность путешествовать, но чаще всего есть места, в которых хотелось бы побывать. Решение данной проблемы можно получить с помощью 3D-реконструкций.

В ходе выполнения данной работы была затронута и изучена некоторая часть такого актуального направления как компьютерное зрение и обработка изображений. Была проведена исследовательская работа и анализ существующих методов построения 3D-моделей.

В ходе реализации одного из методов были решены следующие задачи:

1. Нахождение зависимости взаимного расположения камер при создании изображений и конечного результата;
2. Анализ алгоритмов поиска особых точек;
3. Анализ методик сопоставления ключевых точек в пары с последующей фильтрацией «плохих» пар;
4. Ректификация стереопары;
5. Получение карты глубины.

Полученные в данной работе результаты могут быть использованы в качестве базы для других проектов 3D-реконструкций. Например, для создания онлайн музеев, «посещение» которых создаст полноценное ощущение непосредственного пребывания в соответствующих залах.

Список литературы

1. J. D. BRIERS Laser speckle contrast imaging for measuring blood flow, *Optica Applicata*, 2007, 14 p.
2. R. Hartley, A. Zisserman *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2004. 673 p.
3. Шапиро Л. Компьютерное зрение/ под ред.: С. М. Соколов, пер.: А. А. Богуславский, Л. Шапиро. М.: БИНОМ. Лаборатория знаний, 2013. 762 с.
4. David A. Forsyth, Jean Ponce *Computer Vision: A Modern Approach*, Pearson, 2011. 792 p.
5. Курс *Digital Image Processing*.
<http://freevideolectures.com/Course/2316/Digital-Image-Processing-IIT-Kharagpur>
6. UCF Computer Vision Video Lectures 2012,
https://www.youtube.com/playlist?list=PLd3hlSJxX_Imk_BPmB_H3AQjFKZS9XgZm
7. Курс "Введение в компьютерное зрение", 2015
<https://www.slideshare.net/ktoshik/cv2015-01-introimage>
8. S. M. Seitz and C. R. Dyer, *Proc. Computer Vision and Pattern Recognition Conf. Photorealistic Scene Reconstruction by Voxel Coloring*, 1997, 1067-1073.
9. R. Yang, M. Pollefeys, and G. Welch. *Dealing with Textureless Regions and Specular Highlight: A Progressive Space Carving Scheme Using a Novel Photo-consistency Measure*, *Proc. of the International Conference on Computer Vision*, pp. 576-584, 2003
10. H. Jin, S. Soatto and A. Yezzi. *Multi-view stereo reconstruction of dense shape and complex appearance* *Intl. J. of Computer Vision* 63(3), p. 175-189, 2005.
11. А. Т. Вахитов, Л. С. Гуревич, Д. В. Павленко *Обзор алгоритмов стереозрения*, Санкт-Петербургский государственный университет. 19 с.

12. Тупицын И. В. Реконструкция трехмерной модели объекта на основе стереопары при решении задач 3D-моделирования, Вестник Сибирского государственного аэрокосмического университета им. академика М.Ф. Решетнева. 2011. 5 с.
13. Tuytelaars T., Mikolajczyk K. Local Invariant Feature Detectors: A Survey // Foundation and Trends in Computer Vision. 2007.
14. Brown M., Lowe D.G. Automatic Panoramic Image Stitching using Invariant Features // International Journal of Computer Vision. 2007.
15. Szeliski R. Computer Vision: Algorithms and Applications. Springer, 2010.
16. Ethan Rublee, Vincent Rabaud, Kurt Konolige, Gary Bradski ORB: an efficient alternative to SIFT or SURF. Willow Garage, Menlo Park, California, 2011. 8 p.
17. E. Karami, S. Prasad, M. Shehata Image Matching Using SIFT, SURF, BRIEF and ORB: Performance Comparison for Distorted Images. Faculty of Engineering and Applied Sciences, Memorial University, Canada, 2015. 10 p.
18. Common Interfaces of Descriptor Matchers.
http://docs.opencv.org/2.4/modules/features2d/doc/common_interfaces_of_descriptor_matchers.html
19. Martin A. Fischler and Robert C. Bolles (June 1981). «Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography». *Comm. Of the ACM* 24: pp. 381–395.
20. Kaehler A. Learning OpenCV: Computer Vision in C++ with the OpenCV Library, 2nd edition: O'Reilly, 2013. 575 p.
21. Dawson-Howe K. A Practical Introduction to Computer Vision with OpenCV, John Wiley & Sons Ltd., 2014. 235 p.
22. Baggio Daniel Mastering OpenCV 3. Packt Publishing, 2017. 250 p.
23. Radu Bogdan Rusu and Steve Cousins 3D is here: Point Cloud Library (PCL), Willow Garage 68 Willow Rd., Menlo Park, CA 94025, USA

24. Irene Reisner-Kollmann Reconstruction of 3D Models from Images and Point Clouds with Shape Primitives, Faculty of Informatics at the Vienna University of Technology, 2013. 108 p.
25. Patent SIFT: <https://www.google.com/patents/US6711293>
26. Patent SURF: <http://www.google.com/patents/US20090238460>
27. Веб-сайт OpenCV. <http://opencv.org/>
28. Веб-сайт PCL. <http://pointclouds.org/>